

UiO • Institutt for informatikk

Det matematisk-naturvitenskapelige fakultet

Petter Nielsen

Case: nettbutikk

INF3290 - Store og komplekse informasjonssystemer

30.11.2017



Agenda

- Økosystemer
- Tre historier om nettbutikker med økende grad av kompleksitet med fokus på utfordringer for «utviklerne»
 1. System
 2. Stort og komplekst informasjonssystem
 3. Globalt stort og komplekst informasjonssystem

Læringsmål

- Forstå økosystemperspektivet og hvordan det er relevant i forhold til store og komplekse informasjonssystemer
- Gjenkjenne til tre «myter»
 - Myten om de ensomme informasjonssystemene
 - Myten om de ensomme systemutviklingsprosjektene
 - Myten om de ensomme brukerne
- Kjenne til utfordringer store og komplekse informasjonssystemer skaper

Metaforer

- Hva er en metafor?
 - Å beskrive et fenomen med et begrep/ord som egentlig beskriver noe annet
 - Gir oss innsikt, men har alltid begrensninger (bør ikke tøyes for langt)
- Hvilke metaforer bruker vi i INF3290?
 - Kultivering av installert base
 - Infrastruktur
 - Bootstrapping

Gapet mellom utviklere og brukere (Grudin 1991)

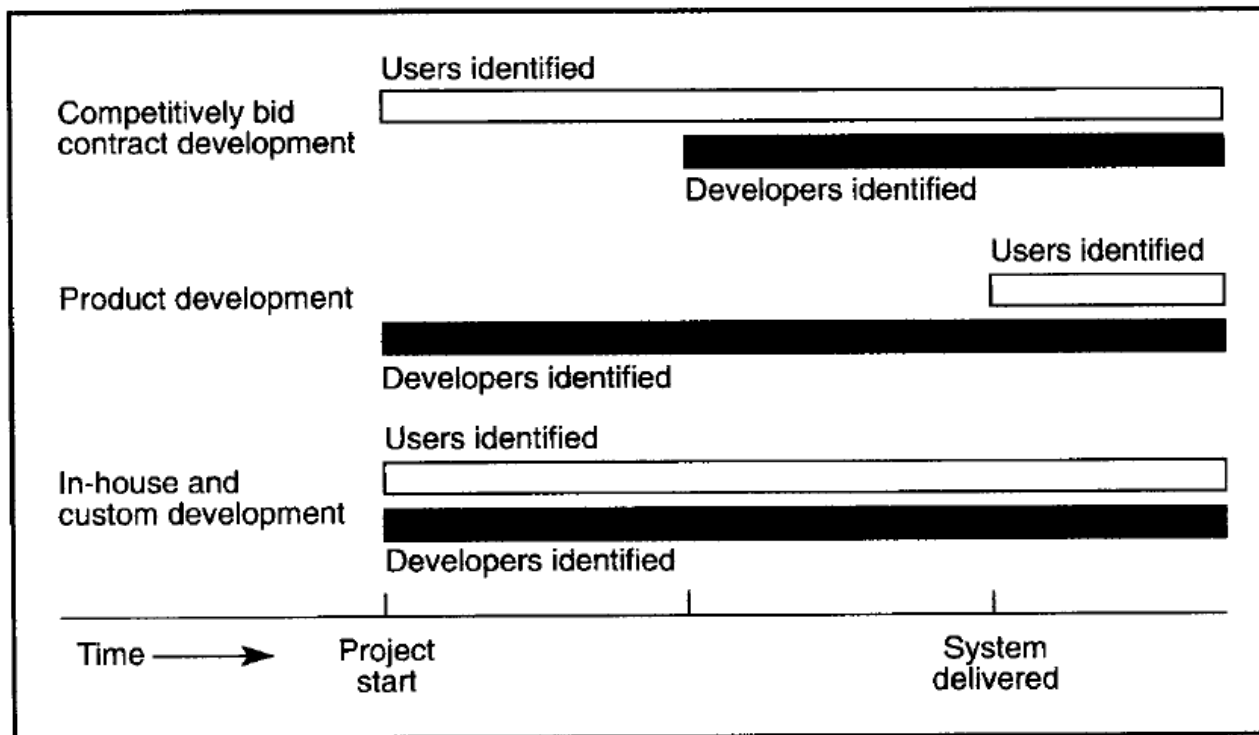
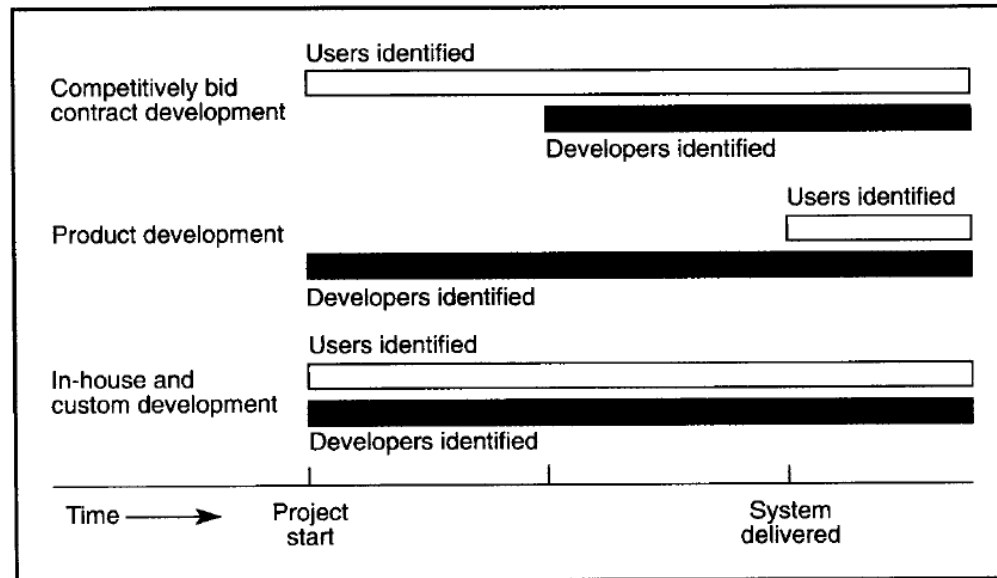


Figure 1. Project time lines with points of user and developer identification.



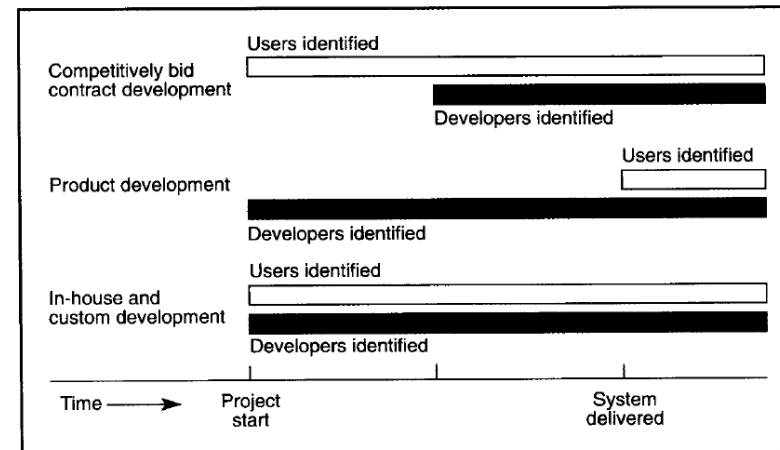
Er Grudin's modell fortsatt relevant?

- Hvilke implikasjoner har store og komplekse informasjonssystemer?
- Er forholdet mellom utvikler og bruker de samme?
- Er tidsdimensjonen fortsatt relevant?



Store og komplekse informasjonssystemer

- Utviklere er brukere og brukere er utviklere: plattformer
- Bygger alltid på noe eksisterende – har ingen ende i tid og rom: økosystemer
- Komplekst forhold mellom brukere og utviklere



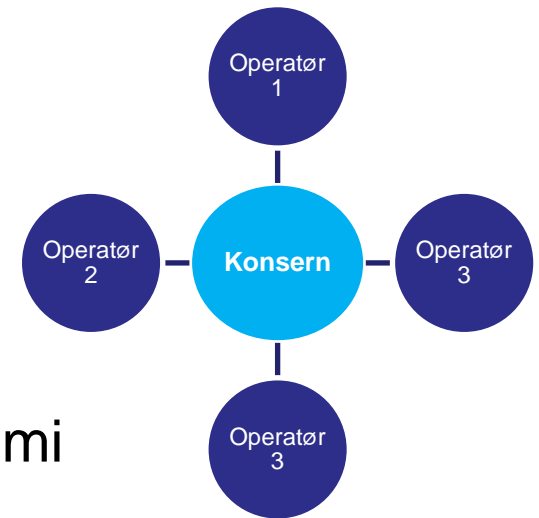
Økosystemer: Definisjon

- Kontekst for systemutvikling (Nielsen 2016):
 - “... a collection of software projects which are developed and evolve together in the same environment ...”
 - “... multiple levels of generification ...”
 - “... a set of actors on top of a common technological platform that results in a number of software solutions or services ...”
 - “... co-evolution of multiple implementation projects ...”
- Implikasjoner for systemutviklere (Dittrich 2014):
 - “Parts of the design and innovation is deferred to other actors closer to the use context.”
 - “Multiple layers of actors customizing and configuring software is involved,
 - “and development is driven by different factors such as bug fixes, the need for new features and technical reengineering.”

Nettbutikker – 3 historier

Hvor kommer historiene/eksemplene fra?

- Teleoperatør med 11 selskaper i 11 forskjellige land (Skandinavia, Sør-Øst Europa og Asia)
- Aktiviteter sentralt i konsernet for å oppnå:
 - Stordriftsfordeler
 - Kortere tid til markedet
 - Læring på tvers
- «Norsk» modell
 - Veldig forskjellige markeder
 - Veldig forskjellig modenhet
 - Operatører med stor grad av autonomi



Økende kompleksitet

System

Development Specification driven

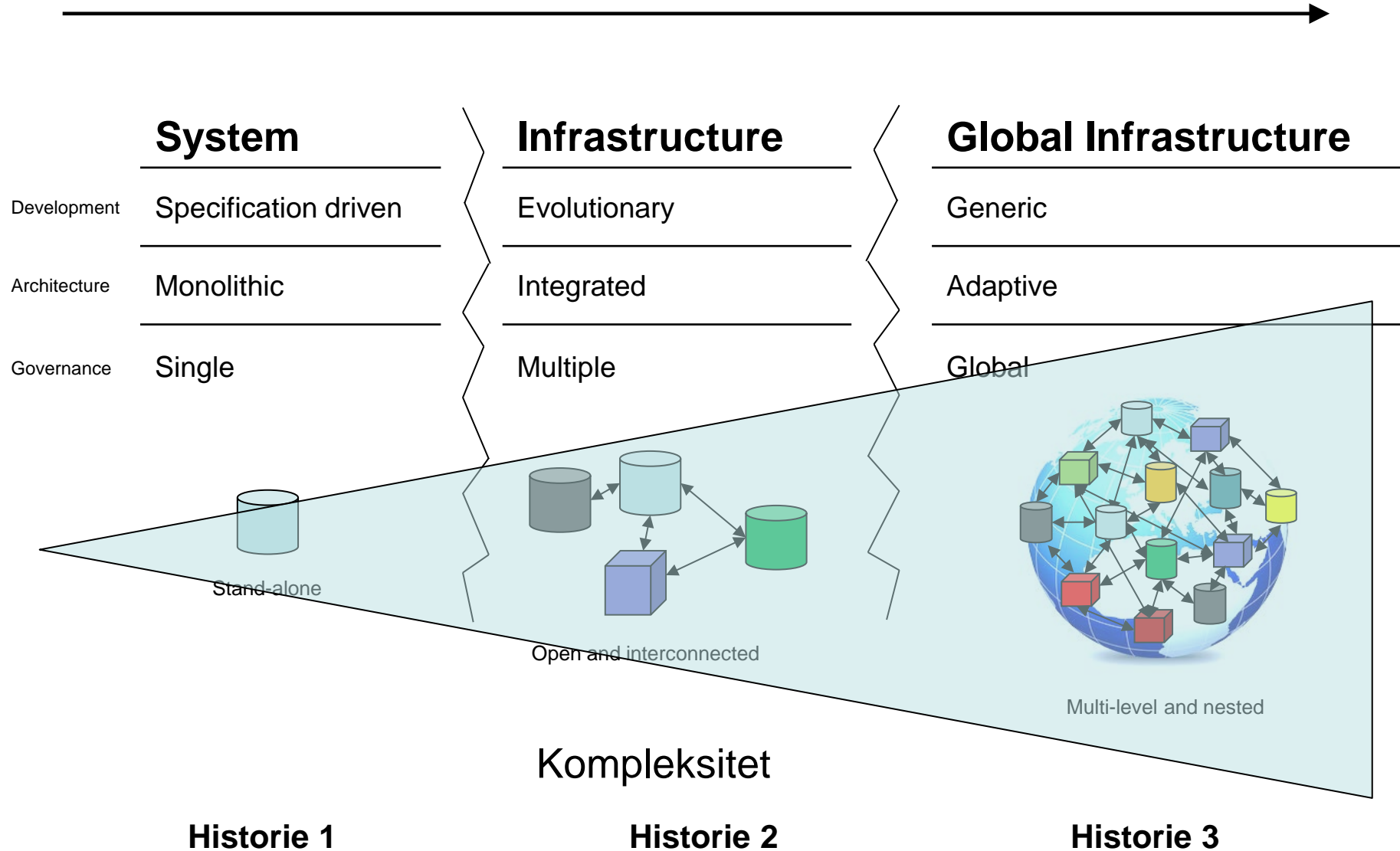
Architecture Monolithic

Governance Single



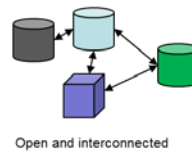
Stand-alone

Tid (historisk) og modenhet for systemer



Historie 1

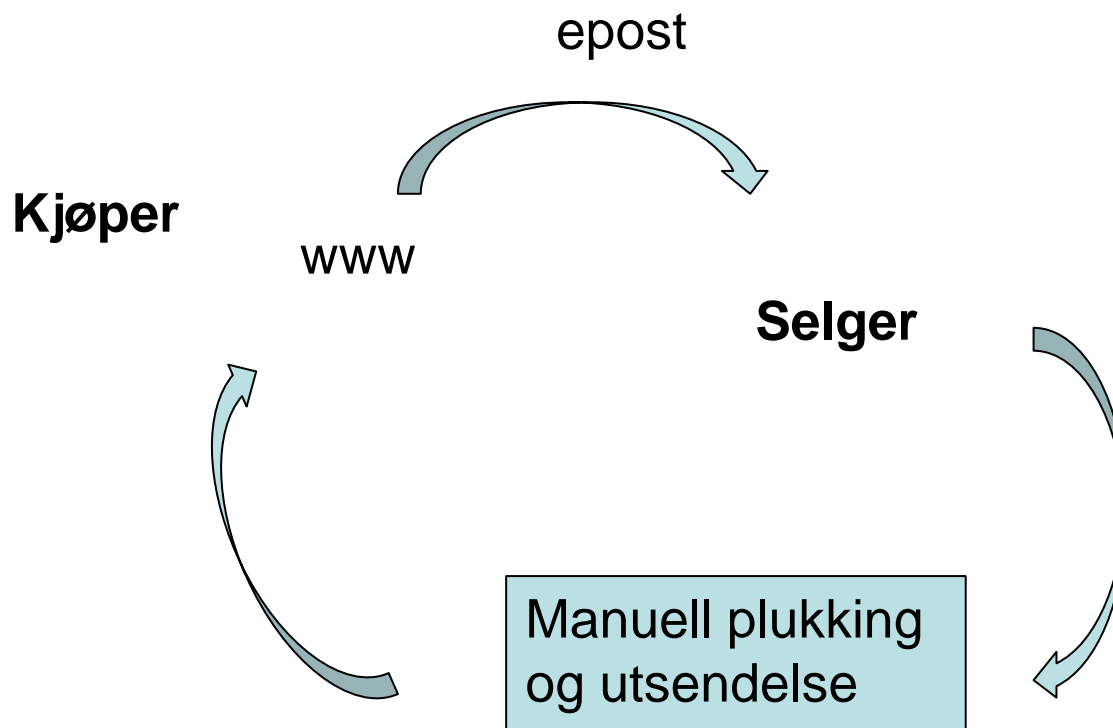
Nettbutikker som System



Hvor langt tid tar det å lage en nettbutikk?

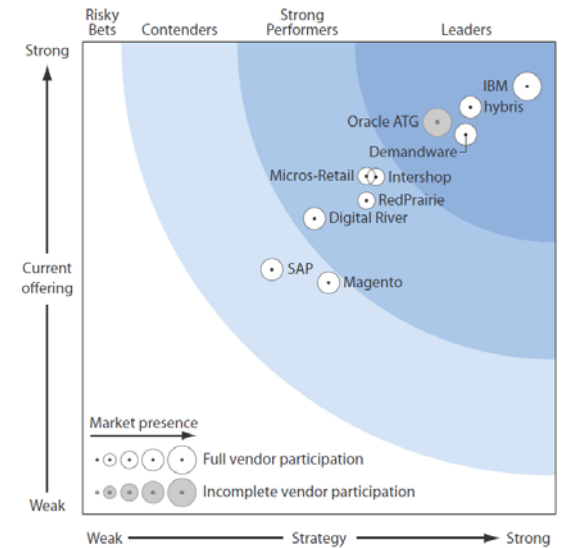
- Ca. 17 minutter

Kompleksitet?



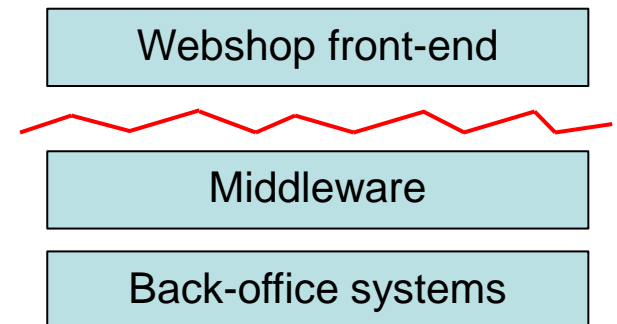
Nettbutikker er enkelt

- Alle kan lage dem
- En enkel handlevogn
- Løsninger er tilgjengelige:
 - Modne COTS plattformer
 - Åpen kildekode (for eksempel Magento)
 - Software as a Service (SaaS) (for eksempel PJ Media)



Eksempel: Teleoperatør i Sør-Europa

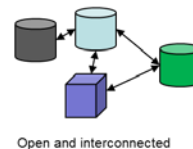
- Lanserte sin nettbutikk kjapt og greit
- Utviklet løsningen innomhus med hjelp av konsulenter
- Sentrale egenskaper
 - + Implementert veldig hurtig
 - + Billig
 - + Men «bare front-end»



Eksempel: Teleoperatør i Sør-Europa

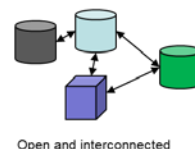
Svakheter

- Manuelt og dobbeltarbeid – for ordrehåndtering og produktdatabase – ikke skalerbar
- Ingen støtte for krysskanal
- Frittstående betyr ingen betalingsløsninger eller personalisering
- Hardkodet (mangel på fleksibilitet)



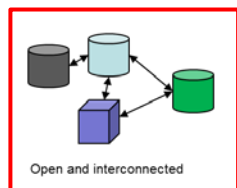
Nettbutikker som System - Oppsummert

- Utfordringer med «primitive» nettbutikker
 - Kundene vil sammenligne med andre nettbutikker - skuffende
 - Frustrerende for de som jobber med butikken og salg
- Bruker ikke mulighetene – møter ikke forventningene
 - Alle kan få til dette
 - Tar ikke ut potensialet i å koble fysiske butikker med nettbutikker



Historie 2

Nettbutikker som stort og komplekst informasjonssystem



Typiske/generiske krav til nettbutikker

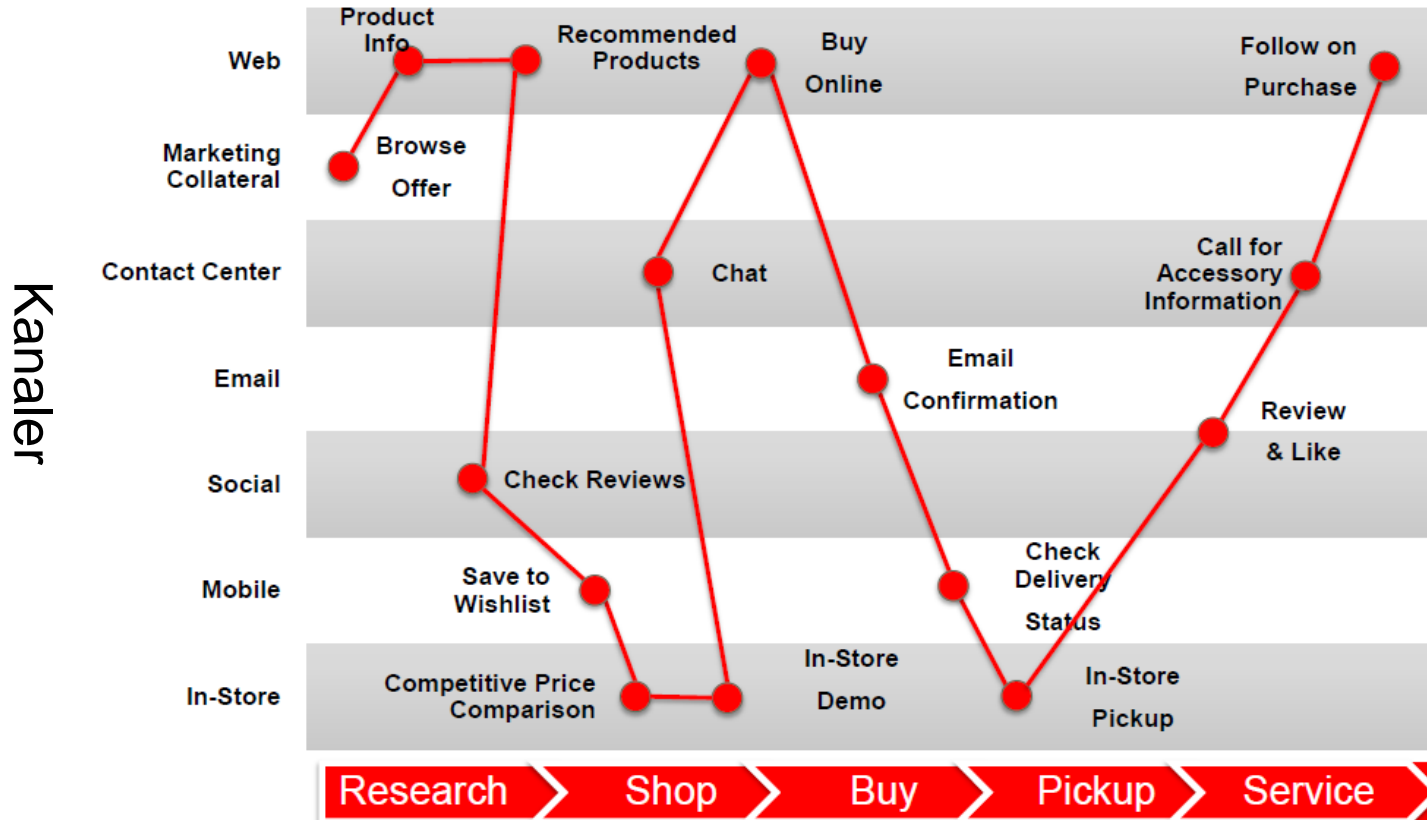
1. Grunnleggende funksjonalitet (handlekurv etc.)
2. Personlige tilbud og individuell prising
 - Identifisering av brukere på web, smarttelefoner og nettbrett
 - Oppfølging av kunder i salgsprosessen i forhold til kunde verdi, tjenesteportefølje og kjøpshistorikk
3. Tett integrert, med for eksempel “mine sider”, kundedatabase etc.
4. Multikanal, f.eks. kjøp på nett og hent i butikk
5. For selgere: fleksibilitet - rask og enkel endring av funksjonalitet, utseende og innhold

Eller?



Kjøpsprosesser og kryss-kanal

Behov for integrasjon av forretningsprosesser

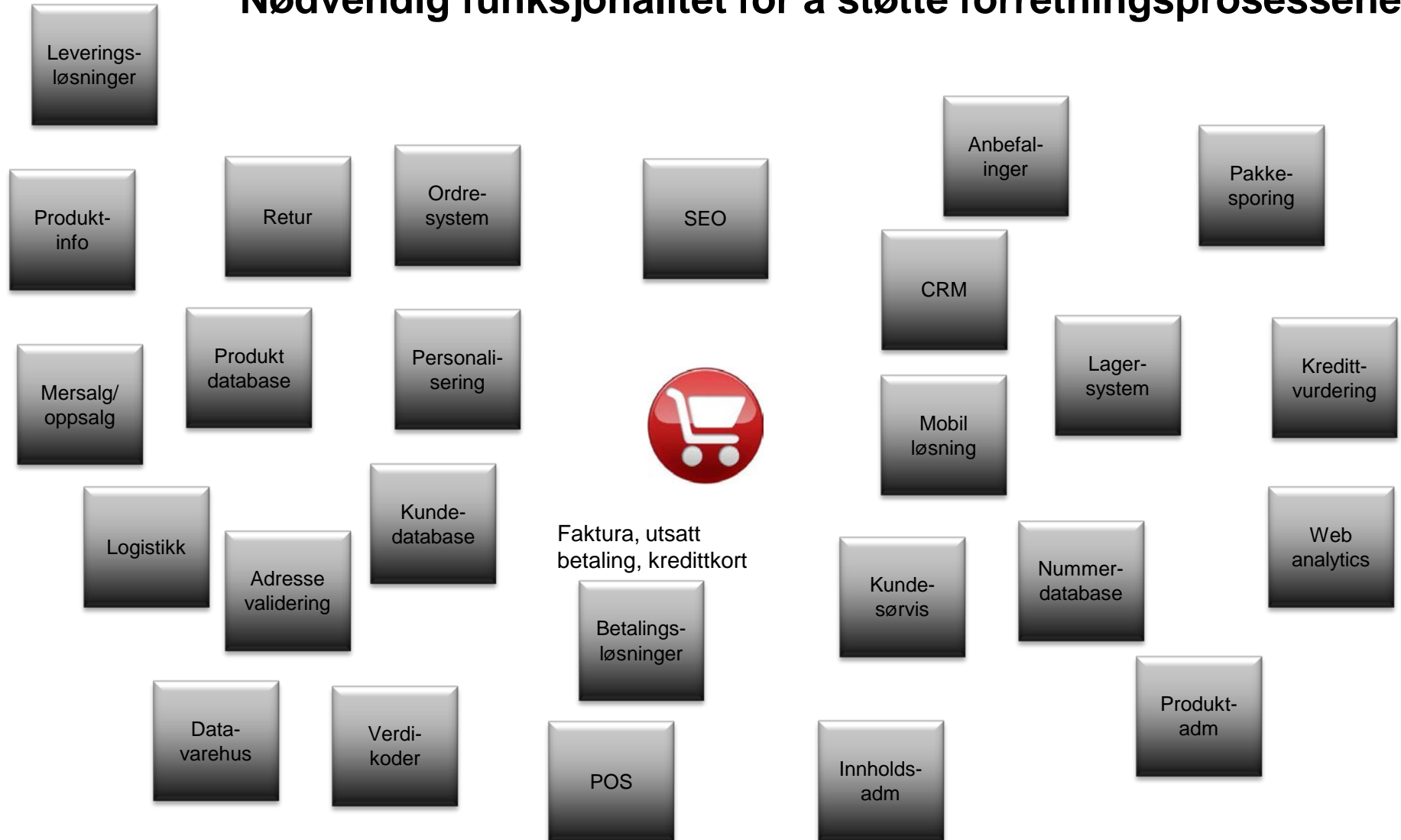


Stadier i kjøpsprosessen

F.eks. Bring (Post i butikk, på døra samme dag etc.

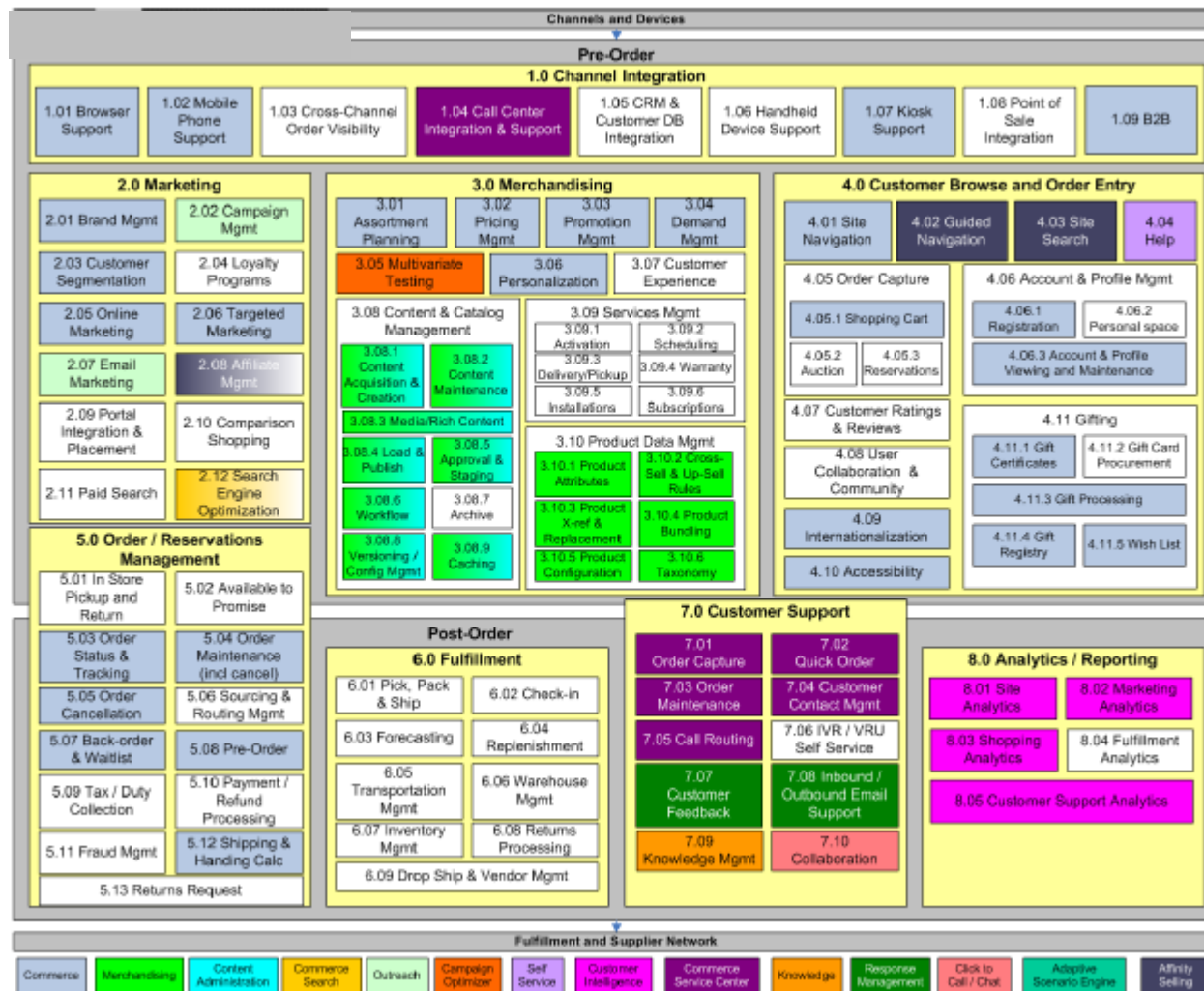
Funksjonell kompleksitet

Nødvendig funksjonalitet for å støtte forretningsprosessene



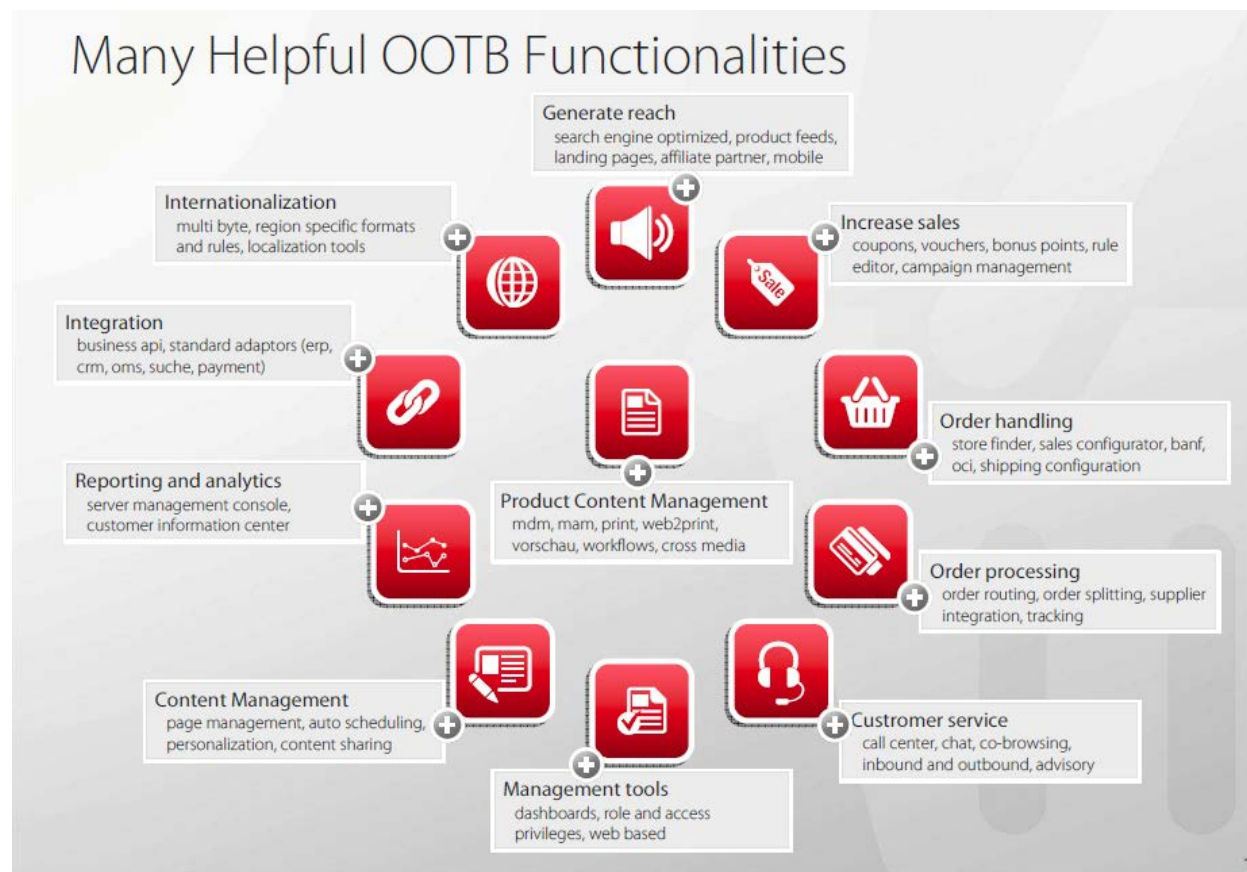
Utfordring Funksjonell arkitektur

Del opp i små komponenter



Funksjonell arkitektur

Eller en plattform?



Utfordring: Krav om omfattende integrasjon

Business Support Systems (BSS)

- CRM (Customer Relationship Management)
- Warehouse (ERP) for inventory management and fulfillment
- Product catalogue
- Analytics
- Business Intelligence

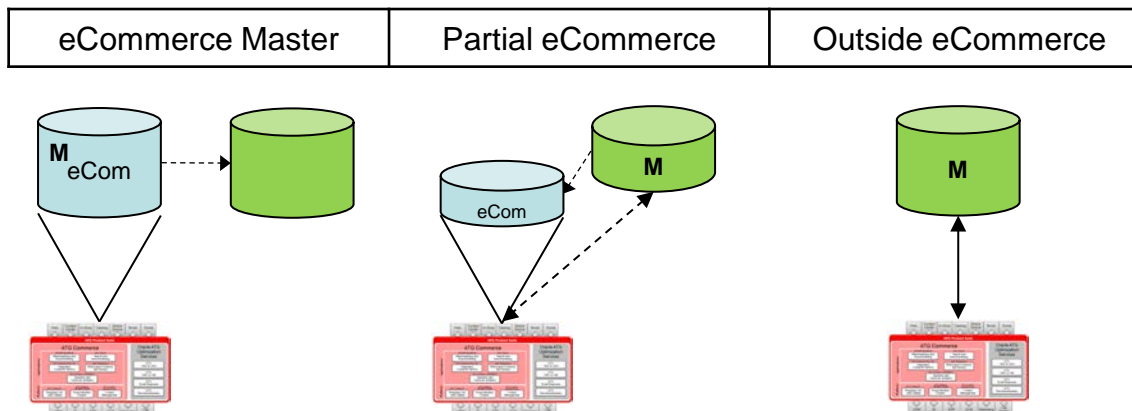
External systems

- Number portability database
- Logistic/ Shipping providers for delivery
- Payment providers
- Address validation
- Credit check
- Geo-location

Ifølge nettbutikkleverandører, integrasjon kan stå for 80% av kostnadene

Utfordring: Hvem styrer produktdata?

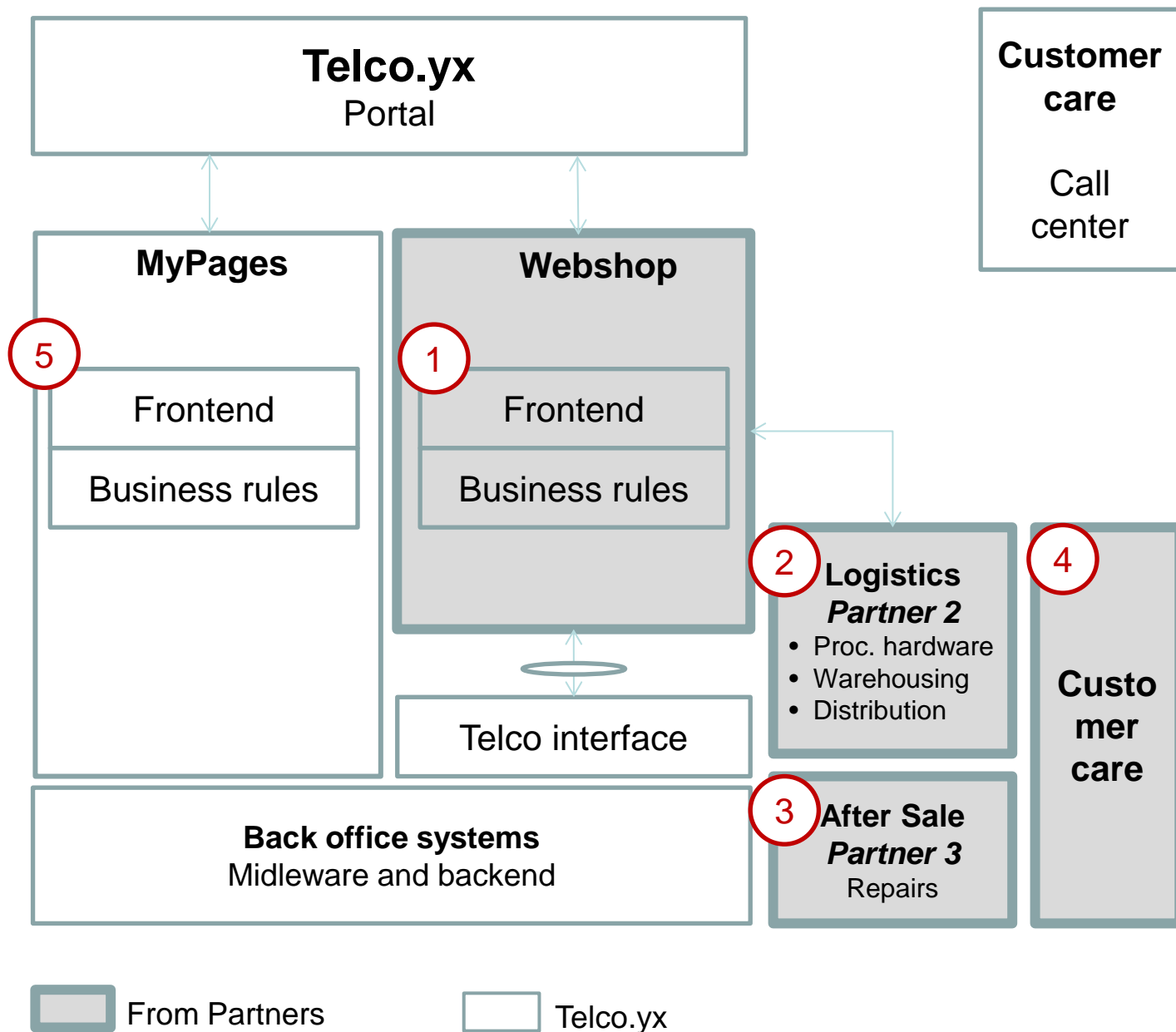
- Nettbutikkdata – relevant bare for nettbutikk (3D bilder)
- Andre data
 - Relevant for mange/alle kanaler slik som nettbutikk, kundesørvis, butikker etc.
 - Globale data må lagres lokalt i nettbutikk for å unngå forsinkelser
 - Hva skjer om systemer som «hoster» produkt master går ned?
- Produktmaster eksempel:

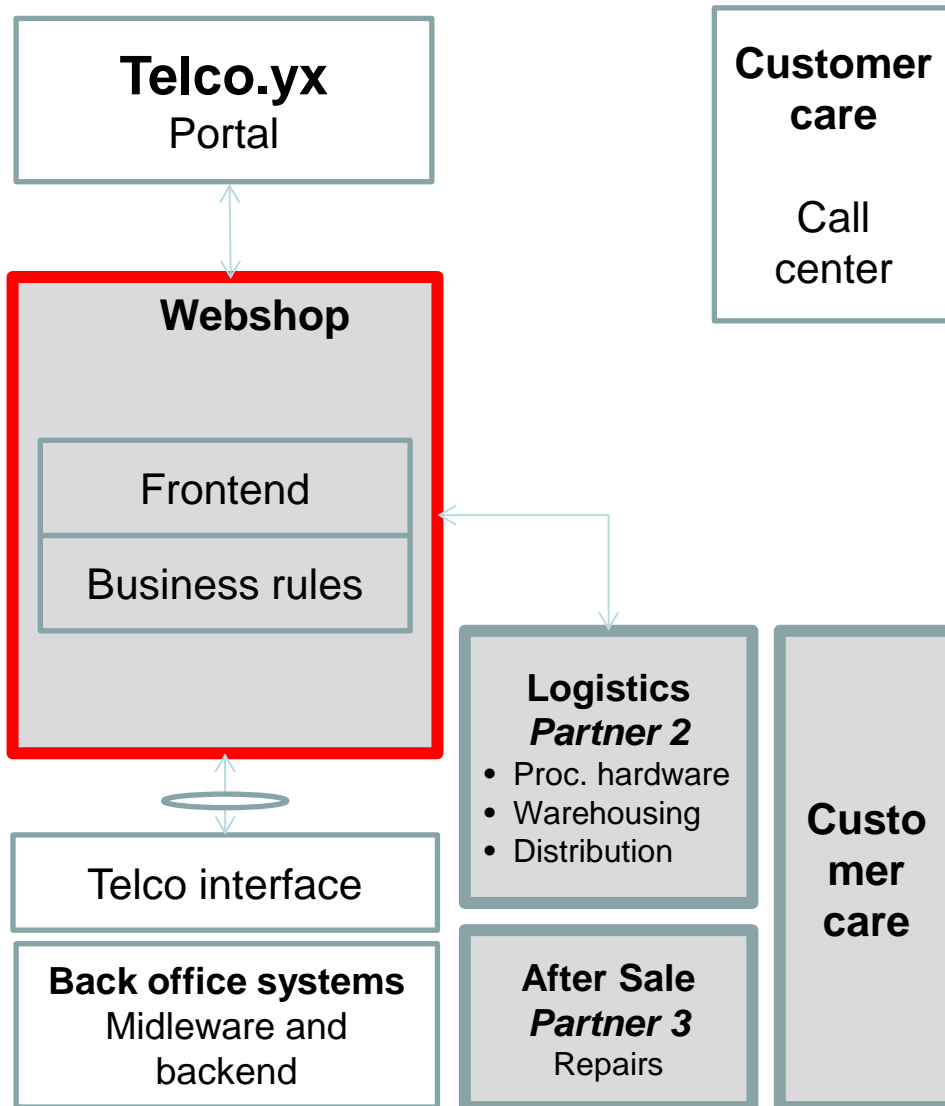


Utfordring: «Arkitektur»

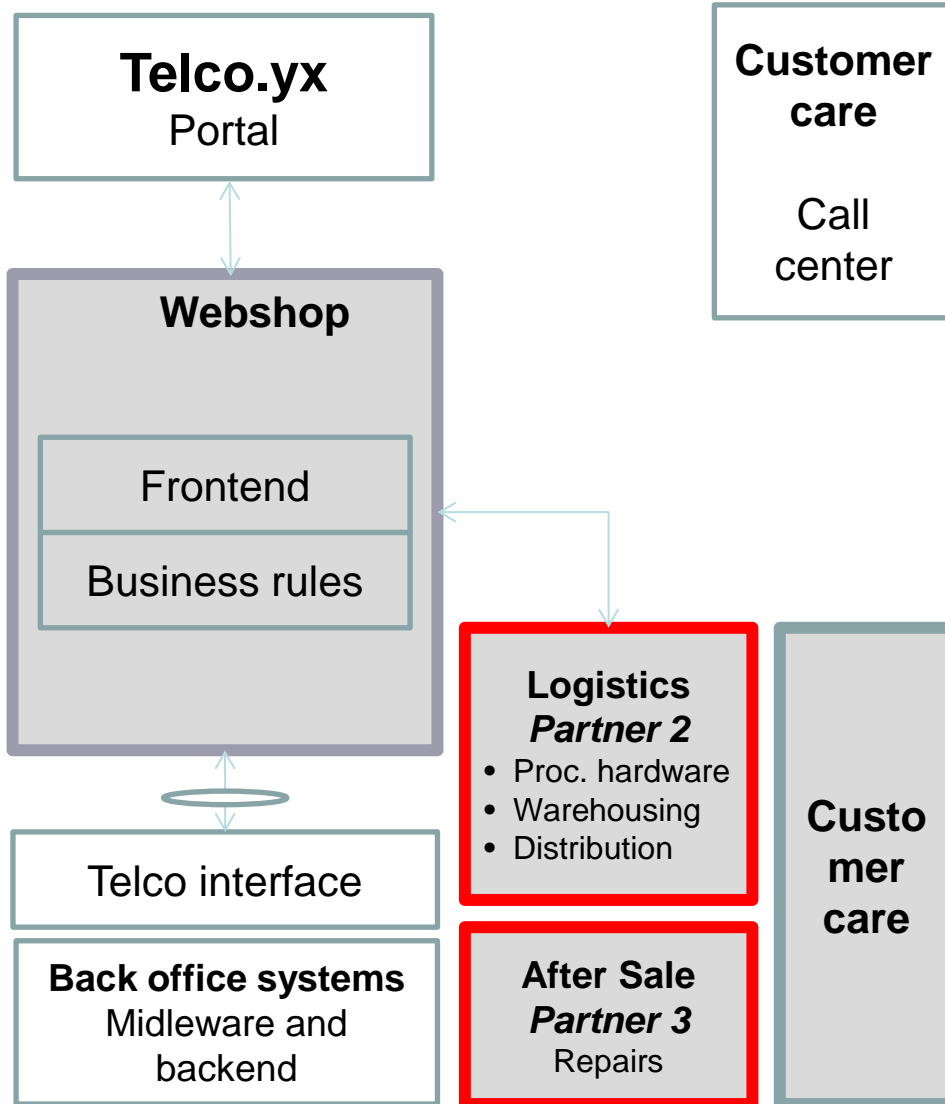
Eksempel: Nettbutikk i Nordisk operatør

1. Nettbutikk
 - Kundegrensesnitt
 - Admin grensesnitt
 - Foretningsregler
2. Logistikk
 - Innkjøp/varehus
3. Ettersalg
 - Reparasjoner
4. Kundestøtte
 - Håndsett
5. Mine sider
 - login, ingen kobling til nettbutikken

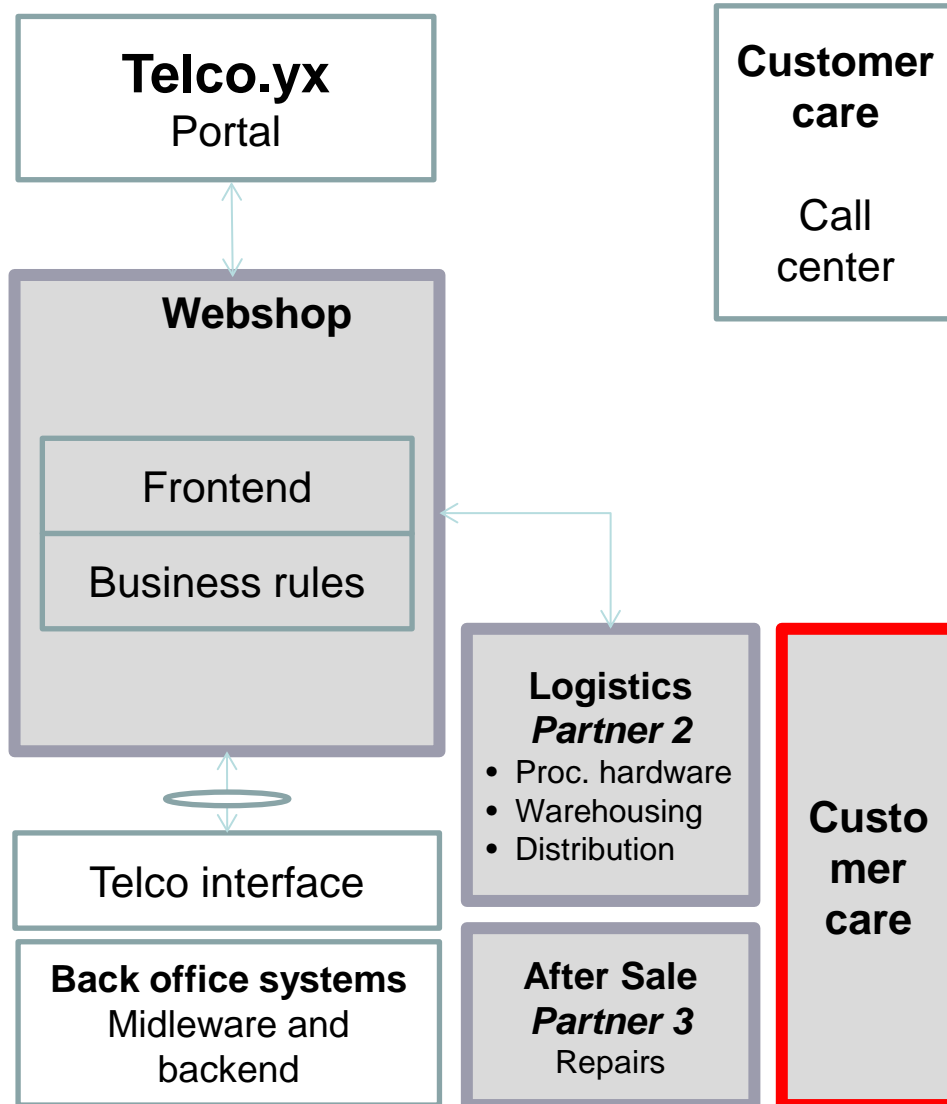




1. Mangler grunnleggende funksjonalitet
2. Ingenting er gratis
3. Lang tid for utvikling, forsinkelser
4. "Hjemmelaget"
5. Partner 1 har begrenset kompetanse, konsulenter gjør jobben
6. Manglende gjennomsiktighet
7. Telco kontrollerer ikke kundereisen
8. Telco mangler kompetanse om løsningen og salg av håndsett generelt



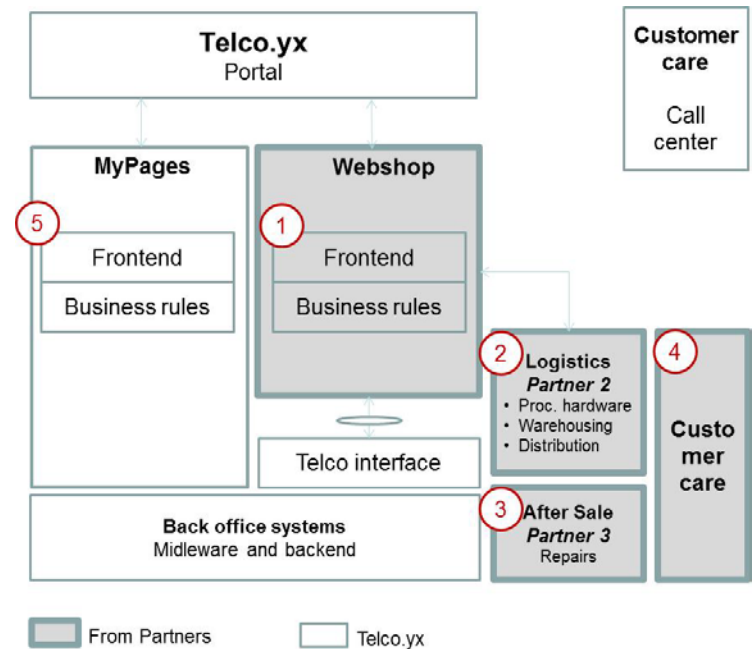
1. Logistikk og etter-salg fungerer fint
2. Men betaler mer enn konkurrentene



1. Kunder må settes over fra Telco til partner, noe som innebærer:
 - Dårlig kundeopplevelse
 - Telco kontrollerer ikke kundeopplevelsen
 - Kan ikke måle kundeopplevelsen

Mulig å endre dette - hva fikk vi til?

- Alle var enige om utfordringene
- Men for stor risiko å gjøre endringer
 - Politisk vanskelig da deleier av partnerne
 - Teknisk løsning kom med partnerskap
 - Usikkerhet rundt andres evne til å integrere
- Enten små endringer, eller en substansiell endring og investering

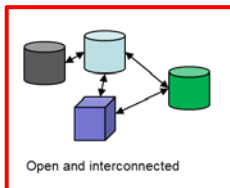


Nettbutikker som stort og komplekst informasjonssystem - Oppsummering

- «Modne» nettbutikker er komplekse
 - Mange forskjellige «domener» og komponenter
 - Mye integrasjon
 - Arkitektur er sentralt – hvilke «komponenter» gjør hva?
 - Verdinettverk med partnere



Stand-alone



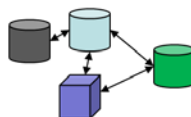
Open and interconnected



Multi-level and nested

Historie 3

Nettbutikker som globalt stort og komplekst informasjonssystem



Eksempel: Nettbutikker i Telco Gruppen

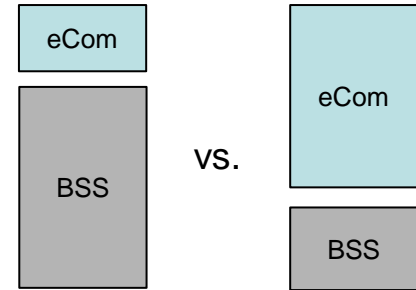
- 11 autonome operasjoner
- 11 forskjellige nettbutikker
 - Mye hjemmesnekret
 - Begrenset funksjonalitet for kunder + mangel på krysskanal integrasjon, delvis manuelt og lite fleksible for salgsavdelingene
- (Delvis) Integrert med 11 forskjellige BSS stakker
- Ingen standardisert ESB, hvis ESB i det hele tatt

Telco Group Strategi: Fra lokale initiativer til industrialisering

- Create top-line growth through being preferred by customers (customer centricity)
- Be a highly cost efficient operator (operational efficiency)
- Operationalized through industrialization/transformation
 - Reduction of complexity and **standardization** of products
 - Best practice sharing of processes, technology and **platforms**
 - Cross border standardization efforts in order to increasingly benefit from **economies of scale** and replication of best practices
- eCommerce industrialization requires global initiatives
 - **Standardization on platforms**
 - Global/regional operating and governance models
 - Coordination with other global initiatives

Standards are the basic for economy of scale

- Common sourcing of platforms (discounts depending on #licenses)
- Coordination of integration (80% is integration)

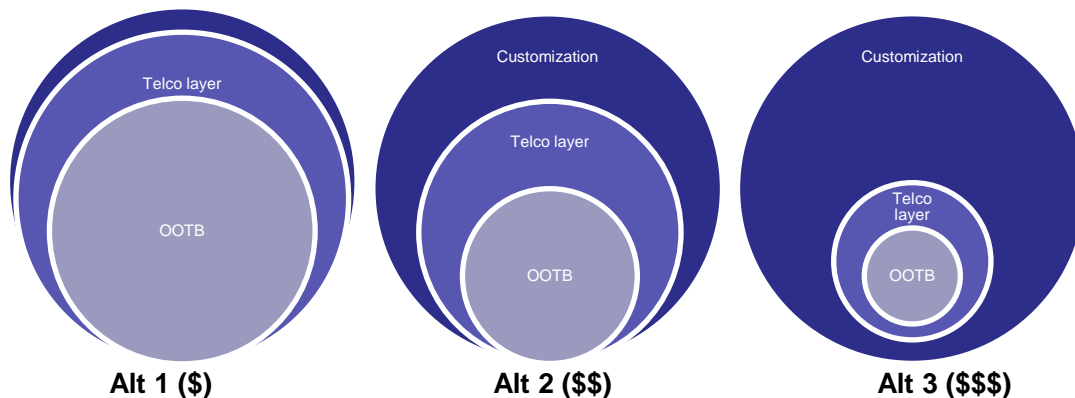


Utfordring 1: Global funksjonalitet

- Mature and immature markets
 - Different transaction volumes
 - Different buying behavior (PC penetration, credit cards etc.)
 - Prepaid mobile for consumer VS. prepaid/postpaid, mobile/fixed/TV for consumer/business
- Mature and immature operations
 - Greenfield or brownfield (in general and eCommerce)
 - Short and long term focus
 - Varying eCommerce investment horizons
 - Thin or fat eCommerce platform needs
- Different strategies and focus
 - Sales or customer care
 - Hunting or farming

Utfordring 2: Global fleksibilitet

- COTS is never OOTB – customization is needed
- The balance between local and global
 - Business Units will (at least on short term) seek to maximize local flexibility
 - Telco Group will seek to minimize local flexibility and maximizing the core for synergies and sustainability
- Assuring buy-in from the operations (living with COTS)

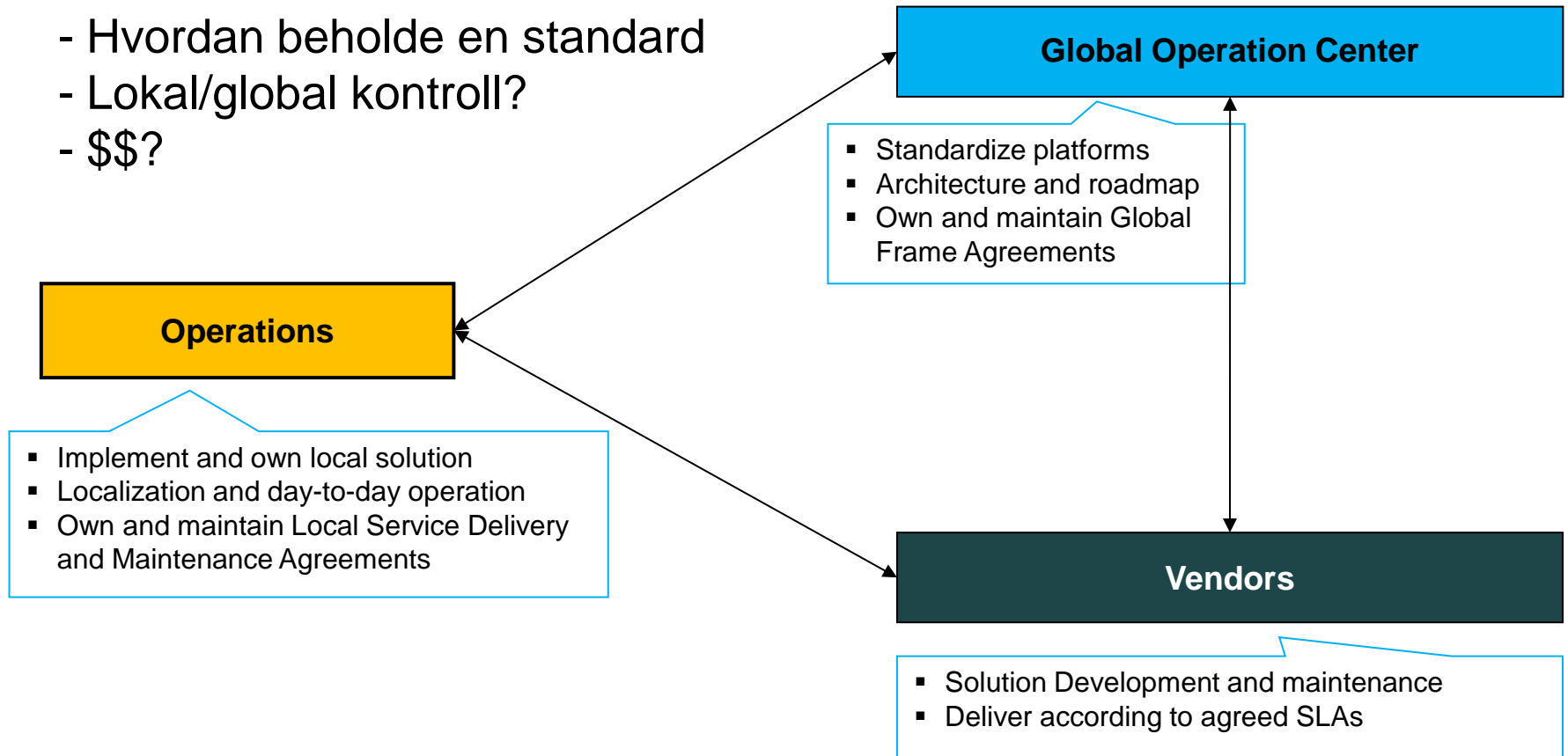


Where to locate functionality?

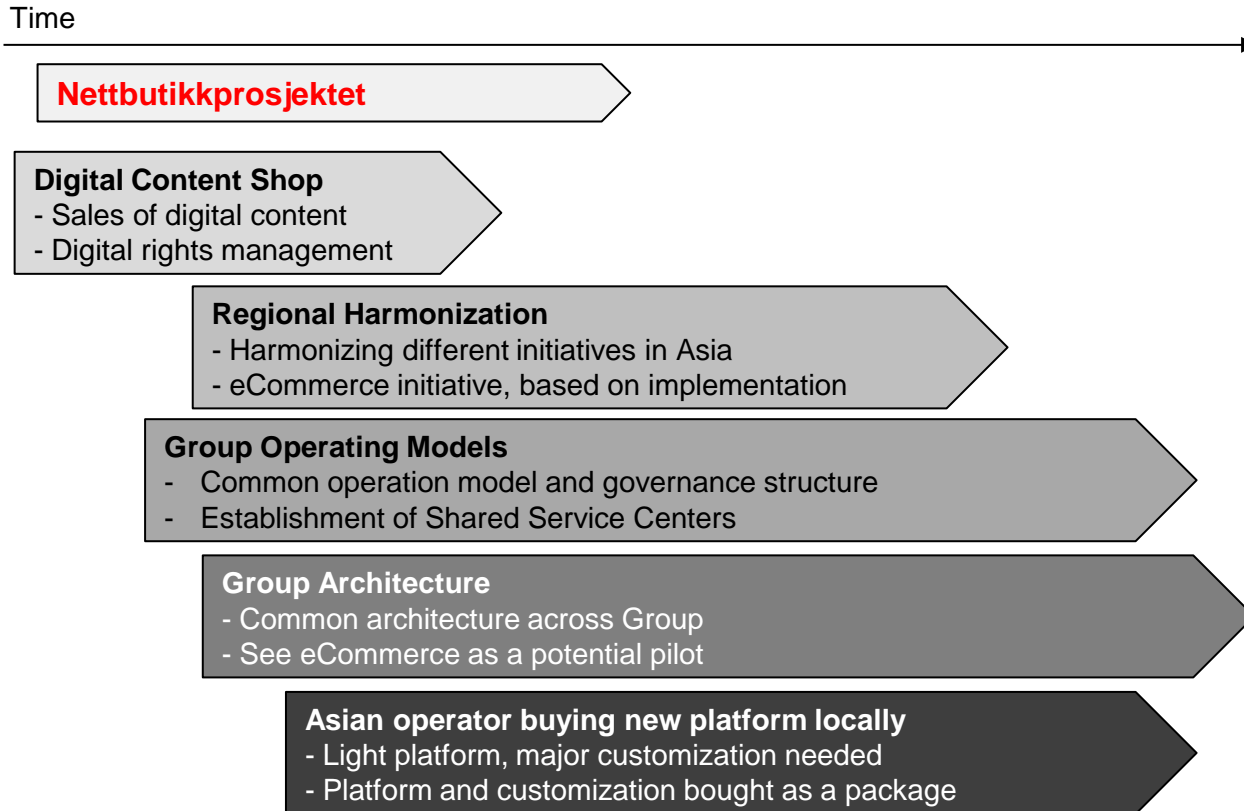
Local Layer	Customization
Standard Layer	Standard across operations
Core Platform	Out-of-the box

Utfordring 3: Styringsmodell

- Hvordan beholde en standard
- Lokal/global kontroll?
- \$\$?

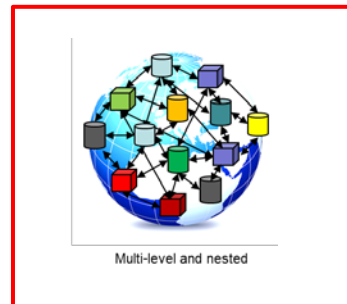
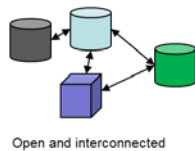


Utfordring 4: Koordinering



Nettbutikker som globalt stort og komplekst informasjonssystem - oppsummering

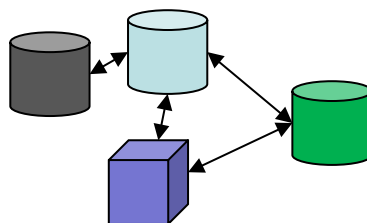
- Hva skal standardiseres?
- Hvor mye skal være globalt vs fleksibilitet?
- Hvordan styre?
- Hvordan relaterte til andre prosjekter?



Økosystem?



Historie 1



Historie 2

Mange aktører
Mange komponenter
Mange avhengigheter
Vanskelig å endre



Historie 3

Mange markeder
Mange prosjekter